ARTIFICIAL INTELLIGENCE

(CSC 462)

ASSIGNMENT # 4

NAME: MUAAZ BIN MUKHTAR

REG NO: FA21-BSE-045

CLASS & SECTION: BSSE-5A

SUBMITTED TO: SIR WAQAS ALI

DATE SUBMITTED: 23-12-2023

Department of Computer Science

QUESTION 1

Given a medical test for a rare disease that is 99% accurate, and the disease occurs in only 0.1% of the population, what is the probability that a person who tests positive actually has the disease?

Answer:

Given:

- Sensitivity of the test (P(Positive | Disease)) = 99% or 0.99
- Prevalence of the disease (P(Disease)) = 0.1% or 0.001
- Specificity of the test (P(Negative | No Disease)) = 99% or 0.99

Now To Find:

P (Disease | Positive), the probability of having the disease given a positive test result.

1. ***P*(Negative | No Disease):**

   = 1−Specificity
   =1−0.99
   =0.01
2. ***P*(No Disease):**

   = 1−P(Disease)
   =1−0.001
   =0.999
3. ***P*(Positive):**

   Use the law of total probability:
   P(Positive)=P(Positive | Disease)×P(Disease)+P(Positive | No Disease)×P(No Disease)
   P(Positive)=0.99×0.001+0.01×0.999
   = 0.01098
4. **Now we will use bayes' Theorem to find *P*(Disease | Positive):**

   P(Disease | Positive)= (P(Positive | Disease)×P(Disease)) / P(Positive)
   P(Disease | Positive)=(0.99×0.001) / (0.99×0.001+0.01×0.999)
   P(Disease | Positive)=0.09

Question No. 2:

A Farmer has a tiger, a goat and a bundle of grass. He is standing at one side of the river with a very week boat which can hold only one of his belongings at a time. His goal is has to take all three of his belongings to the other side. The constraint is that the farmer cannot leave either goat and tiger, or goat and grass, at any side of the river unattended because one of them will eat the other. Using the simple POP algorithm we studied in the lecture, solve this problem. Show all the intermediate and final plans step by step.

Answer:

We'll represent the state of the problem with the following variables:

- *G* for the goat,
- *T* for the tiger,
- *B* for the bundle of grass,
- *F*1 for the farmer on the starting side,
- *F*2 for the farmer on the other side.
- | represents sides of river

The initial state is ***G, T, B, F*1** on one side and ***F*2** on the other side of the river.
The goal state is to have all of them on the other side. The legal operators are moving the farmer with one of the belongings from one side to the other.

<u>Following is the plan step by step to solve this problem:</u>

Initial State:
G, *T*, *B*, *F*1 **|** *F*2

Step 1:
Move the Farmer (*F*) with the Goat (*G*) to the other side:
T, *B*, *F*1 **|** *F*2, *G*

Step 2:
Move the Farmer back to the starting side:
T, *B* **|** *F*2, *G*, *F*1

Step 3:
Move the Farmer (*F*) with the Bundle of Grass (*B*) to the other side:
T **|** *F*2, *G*, *F*1, *B*

Step 4:
Move the Farmer back to the starting side:
T **|** *F*2, *G*, *B*, *F*1

Step 5:
Move the Farmer (*F*) with the Tiger (*T*) to the other side:
| *F*2, *G*, *B*, *F*1, *T*

Step 6:
Move the Farmer back to the starting side:
T **|** *F*2, *G*, *B*, *F*1

Step 7:
Move the Farmer (*F*) with the Goat (*G*) to the other side:
| *F*2, *B*, *F*1, *T*, *G*

Final State:
| *F*2, *B*, *F*1, *T*, *G*

Now, all the belongings are on the other side of the river, and the farmer successfully transported them without any conflict. This plan ensures that at no point is the tiger left alone with the goat or the goat left alone with the bundle of grass.

Question No. 3:

A robot has three slots available to put the blocks A, B, C. The blocks are initially placed at slot 1, one upon the other (A placed on B placed on C) and it's goal is to move all three to slot 3 in the same order. The constraint to this robot is that it can only move one block from any slot to any other slot, and it can only pick the top most block from a slot to move. Using the simple POP algorithm we studied in the lecture, solve this problem. Show all the intermediate and final plans step by step.

Answer:

We'll represent the state of the problem using variables A, B, and C to denote the blocks, and $S1$, $S2$, and $S3$ to represent the slots.
The initial state is A on $S1$, B on top of A, and C on top of B.
The goal is to have A, B, and C in order on $S3$.

Initial State:
A, *B*, *C* |
Step 1:
Move C to $S3$:
A, *B* | *C*
Step 2:
Move B to $S2$:
A | *C*, *B*
Step 3:
Move C to $S2$:
A | *B*, *C*
Step 4:
Move A to $S3$:
| *B*, *C*, *A*
Step 5:
Move C to $S1$:
C | *B*, *A*
Step 6:
Move B to $S3$:
C | *A* | *B*
Step 7:
Move A to $S2$:
| *A*, *C*, *B*
Step 8:
Move C to $S3$:
| *A*, *B*, *C*
Final State:
| *A*, *B*, *C*
Now, all three blocks are on in the correct order. Each step adheres to the constraint that only the topmost block from a slot can be moved, ensuring that the order is maintained.